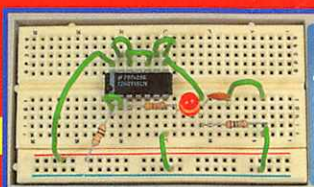
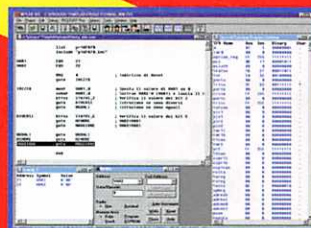
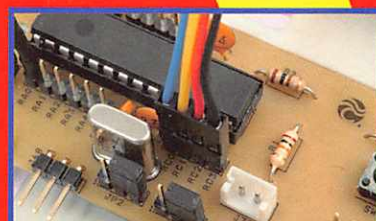
...e costruisci il tuo LABORATORIO DIGITALE

6,90 €

27

HARDWARE

DIGITALE DI BASE

MICROCONTROLLER

DIGITALE AVANZATO

TOTALMENTE PROGRAMMABILE!!!

9 771824 363008 41027

Peruzzo & C.

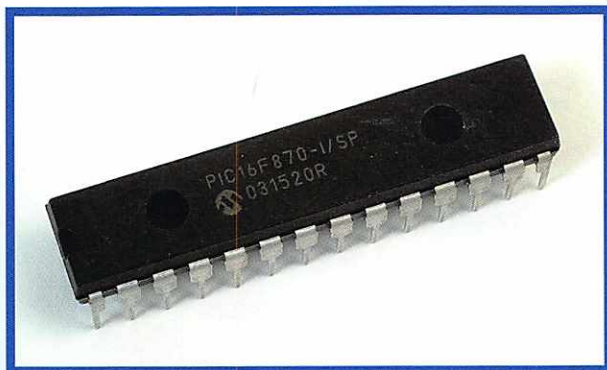
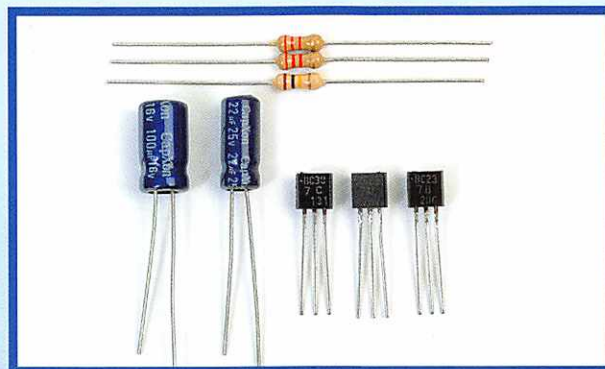
Direttore responsabile:
ALBERTO PERUZZO
Direttore Grandi Opere:
GIORGIO VERCELLINI
Consulenza tecnica e traduzioni:
CONSULCOMP S.n.c.
Pianificazione tecnica
LEONARDO PITTON

Direzione, Redazione, Amministrazione: viale Ercole Marelli 165, Tel. 02/242021, 20099 Sesto San Giovanni (Mi). Pubblicazione settimanale. Registrazione del Tribunale di Monza n. 1738 del 26/05/2004. Spedizione in abbonamento postale gr. II/70; autorizzazione delle Poste di Milano n. 163464 del 13/2/1963. Stampa: Grafiche Porpora s.r.l., Cernusco S/N (MI). Distribuzione SO.DI.P. S.p.A., Cinisello Balsamo (MI).

"ELETTRONICA DIGITALE"
si compone di
70 fascicoli settimanali
da suddividere
in 2 raccoglitori.

RICHIESTA DI NUMERI ARRETRATI. Per ulteriori informazioni, telefonare dal lunedì al venerdì ore 9.30-12.30 all'ufficio arretrati tel. 02/242021. Se vi mancano dei fascicoli o dei raccoglitori per completare l'opera, e non li trovate presso il vostro edicolante, potrete riceverli a domicilio rivolgendovi direttamente alla casa editrice. Basterà compilare e spedire un bollettino di conto corrente postale a PERUZZO & C. s.r.l., Ufficio Arretrati, viale Marelli 165, 20099 Sesto San Giovanni (MI). Il nostro numero di c/c postale è 42980201. L'importo da versare sarà pari al prezzo dei fascicoli o dei raccoglitori richiesti, più le spese di spedizione € 3,10 per pacco. Qualora il numero dei fascicoli o dei raccoglitori sia tale da superare il prezzo globale di € 25,82 e non superiore a € 51,65, l'invio avverrà per pacco assicurato e le spese di spedizione ammonteranno a € 6,20. La spesa sarà di € 9,81 da € 51,65 a € 103,29; di € 12,39 da € 103,29 a € 154,94; di € 14,98 da € 154,94 a € 206,58; di € 16,53 da € 206,58 in su. Attenzione: ai fascicoli arretrati, trascorse dodici settimane dalla loro distribuzione in edicola, viene applicato un sovrapprezzo di € 0,52, che andrà pertanto aggiunto all'importo da pagare. Non vengono effettuate spedizioni contrassegno. Gli arretrati di fascicoli e raccoglitori saranno disponibili per un anno dal completamento dell'opera. IMPORTANTE: è assolutamente necessario specificare sul bollettino di c/c postale, nello spazio riservato alla causale del versamento, il titolo dell'opera nonché il numero dei fascicoli e dei raccoglitori che volete ricevere.

IN REGALO in questo fascicolo

1 PIC 16F870 programmato

IN REGALO nel prossimo fascicolo

1 Transistor BC307
2 Transistor BC237
1 Condensatore elettrolitico da 100 µF
1 Condensatore elettrolitico da 22 µF
2 Resistenze da 2K2 5% 1/4 W
1 Resistenza da 100K 5% 1/4 W

COME RACCOGLIERE E SUDDIVIDERE L'OPERA NELLE 4 SEZIONI

L'Opera è composta da 4 sezioni identificabili dalle fasce colorate, come indicato sotto. Le schede di ciascun fascicolo andranno suddivise nelle sezioni indicate e raccolte nell'apposito raccoglitore, che troverai presto in edicola. Per il momento, ti consigliamo di suddividere le sezioni in altrettante cartellette, in attesa di poterle collocare nel raccoglitore. A prima vista, alcuni numeri di pagina ti potranno sembrare ripetuti o sbagliati. Non è così: ciascuno fa parte di sezioni differenti e rispecchia l'ordine secondo cui raccogliere le schede.
Per eventuali domande di tipo tecnico scrivere al seguente indirizzo e-mail:
elettronicadigitale@microrobots.it

Hardware Montaggio e prove del laboratorio

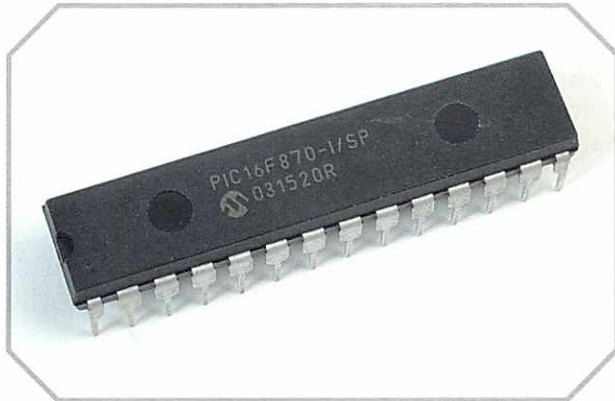
Digitale di base Esercizi con i circuiti digitali

Digitale avanzato Esercizi con i circuiti sequenziali

Microcontroller Esercizi con i microcontroller

Il PIC programmato

PIC 16F870 programmato.

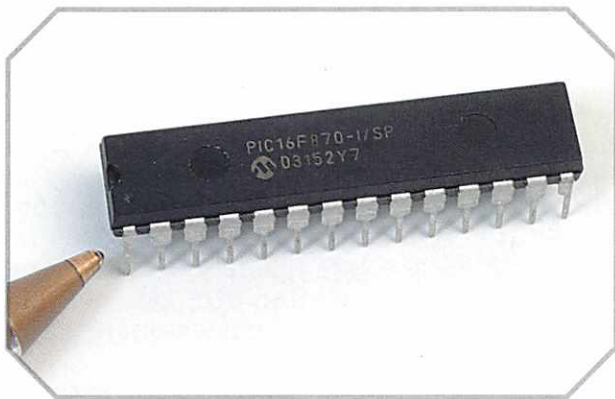
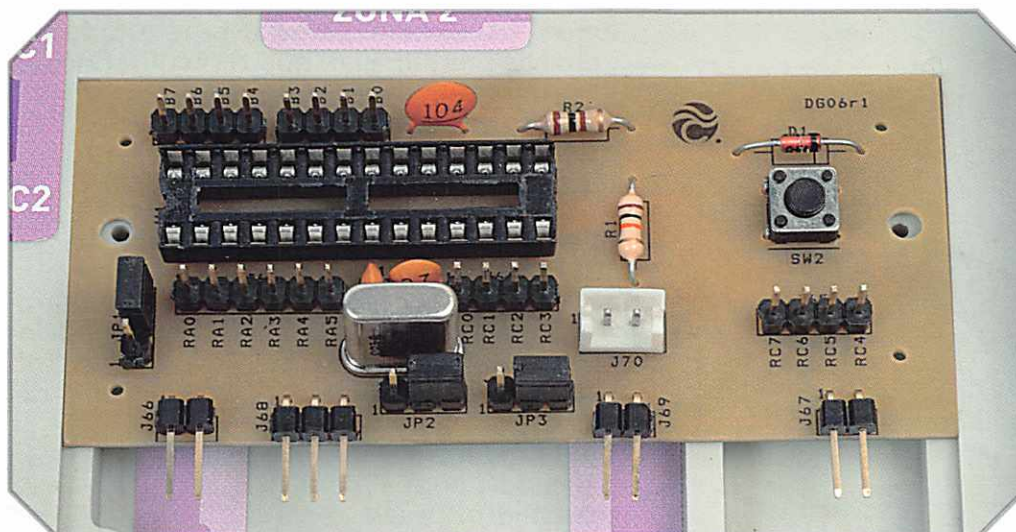
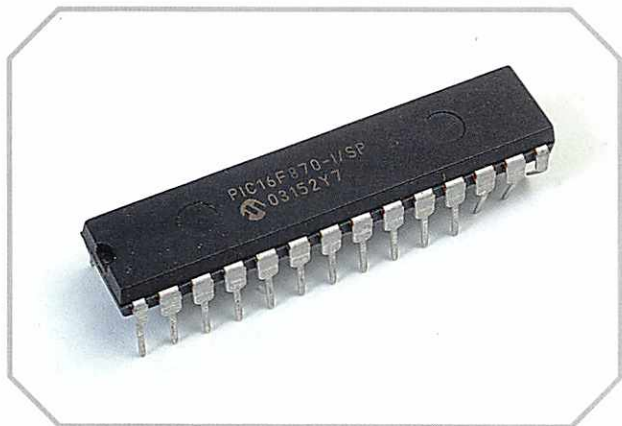
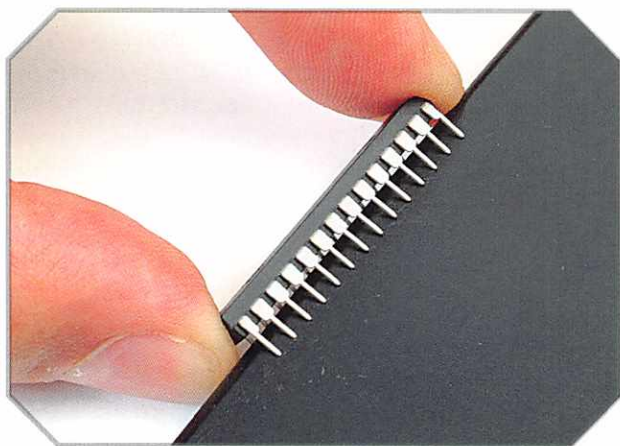
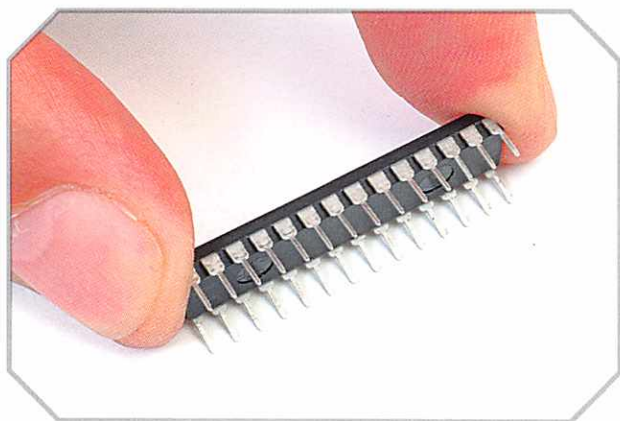
Identificazione del terminale 1.

Con questo fascicolo viene fornito il PIC 16F870 programmato. Contiene un programma dimostrativo con cui si potranno realizzare alcuni esercizi prima di completare il sistema di scrittura.

Alimentazione a 9 V

In questo fascicolo vi spiegheremo anche come collegare il secondo portabatterie alla scheda DG09 di distribuzione dell'alimentazione, allo scopo di poter disporre della tensione da 9 V che sarà utilizzata in alcune prove.

Il PIC

Il PIC 16F870 è fornito in un formato classico, lo standard DIL (Dual In Line) da 28 pin, distribuiti su due file da 14. Per maneggiare questo circuito si devono seguire le precauzioni generali per i circuiti integrati, in quanto sono sensibili alle cariche elettrostatiche: non si devono mai toccare i terminali direttamente. Nell'ambiente industriale esistono elementi per proteggersi dalle scariche elettrostatiche quali polsiere conduttive con presa di terra. Tuttavia in una normale abitazione solitamente non si dispone di questo tipo di accessori.

È necessario prendere alcune precauzioni che di solito danno buoni risultati ed eliminano, o perlomeno riducono, in gran parte i

Scheda DG06 prima di inserire il PIC.

Il PIC potrebbe avere qualche terminale un po' piegato.

Uno dei possibili modi di aprire i terminali.

Metodo per chiudere i terminali e renderli paralleli.

danni prodotti dall'elettricità elettrostatica. Prima di toccare il PIC dobbiamo toccare con la mano diverse zone del tavolo dove lavoreremo e del laboratorio stesso, in maniera che, se avessimo accumulato elettricità statica, molto probabilmente la scaricheremmo. Un'altra misura importante è afferrare il PIC da entrambi i lati evitando di toccare i suoi terminali e, prima di montarlo al suo posto, toccare con un dito la superficie dove verrà appoggiato.

Allineamento dei terminali

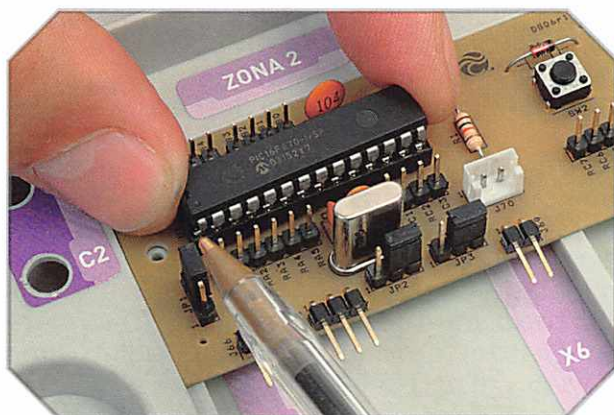
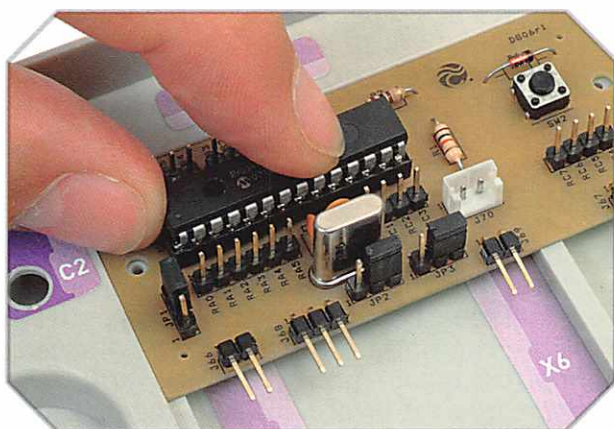
Nonostante il PIC sia fornito con una adeguata protezione dei suoi terminali, può succedere che questi patiscano qualche deformazione. D'altra parte l'elevato numero di pin e la necessità di inserirli contemporaneamente sullo zoccolo, richiede che ogni pin sia ben allineato, con il corrispondente terminale femmina dello zoccolo.

Osservando il PIC potremo vedere che normalmente i terminali risultano un po' più aperti di quanto effettivamente richiede lo zoccolo, in questo caso è sufficiente appoggiare tutti i terminali dello stesso lato, su una superficie dura, e fare una leggera pressione sul contenitore dell'integrato, per piegarli contemporaneamente e perfezionare l'allineamento.

È possibile anche fare l'operazione in senso contrario nel caso fosse necessario, utilizzando in questo caso un paio di pinzette a punta. Queste sono operazioni piuttosto comuni, infatti è abbastanza normale che i terminali degli integrati subiscano qualche deformazione lungo il trasporto. Bisogna tenere presente però che i terminali di questi circuiti sono delicati e tutte queste operazioni devono essere eseguite con attenzione, in quanto muovendo verso un lato o verso l'altro diverse volte il terminale corriamo il rischio di romperlo.

Identificazione del terminale 1

La forma del contenitore del circuito integrato permette il suo inserimento in due posizioni, ma solamente una è corretta. È necessario identificare per prima cosa il terminale 1 del PIC, che è situato di fronte al 28.

Il terminale numero 1 si può identificare in due modi , normalmente tra i terminali 1 e 28 troviamo oltre alla tacca di riferimento dell'integrato anche un piccolo cerchio, inoltre

Prima di inserire il PIC bisogna verificare dove si trova il terminale 1.

Il PIC si inserisce a pressione.

sulla serigrafia della scheda è disegnata la forma dell'integrato e quella della sua tacca. Sulla scheda del circuito stampato il PIC è identificato come U1.

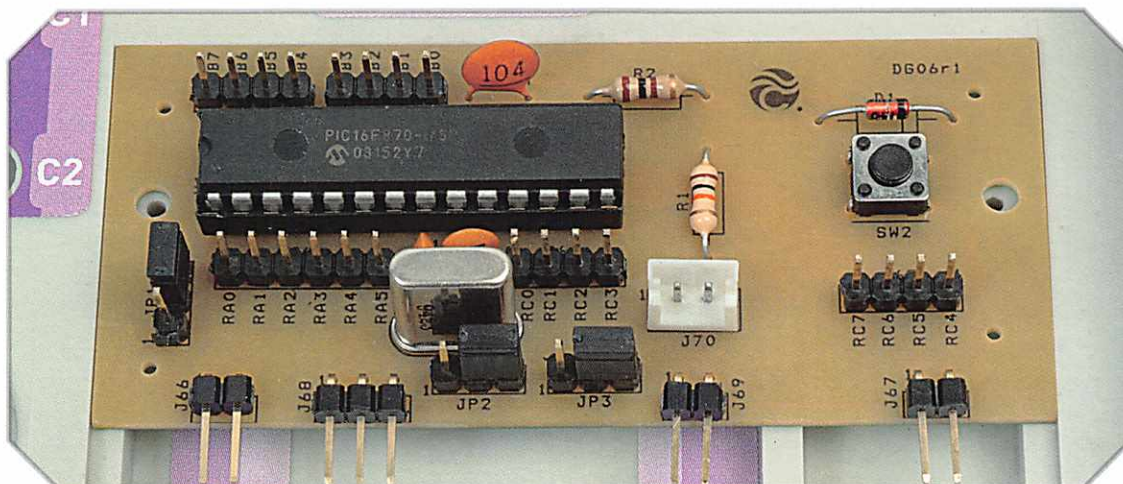
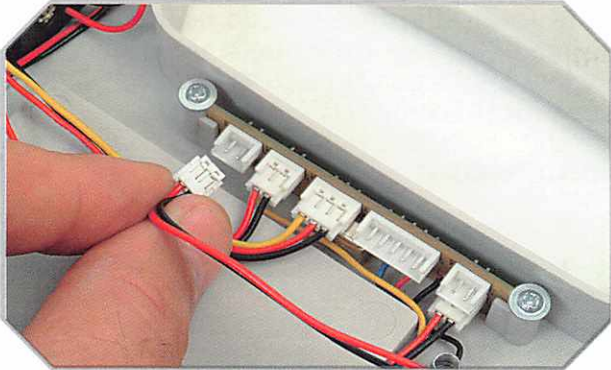
Inserzione

Il PIC si inserisce come tutti gli integrati con questo tipo di contenitore, è sufficiente allineare bene tutti i terminali con quelli dello zoccolo, e fare una leggera pressione fino a quando non sarà bene inserito, quindi ben collegato; verificheremo che nessuno dei suoi terminali sia spostato lateralmente. Infine bisogna verificare che il terminale 1 sia al suo posto, e che quindi l'integrato sia correttamente inserito.

Estrazione

Se per qualche motivo bisogna togliere l'integrato dallo zoccolo, è necessario sollevare i suoi lati poco a poco, utilizzando la punta di un cacciavite a taglio inserendolo fra il contenitore dell'integrato e lo zoccolo, facendo leva con attenzione e passandolo da un lato all'altro sino a quando l'integrato non uscirà dalla propria sede. Questa operazione non deve essere fatta direttamente con le dita, visto la facilità con cui si possono danneggiare i terminali dell'integrato, inoltre potremmo rischiare di pizzicarci le dita con i terminali stessi.

Alimentazione

Questa scheda si alimenta a 5 V, non si può alimentare a 9 V. Il connettore di alimentazione

Circuito stampato DG06 completo con il PIC inserito.

Il secondo portabatterie si collega a J94 della scheda DG09.

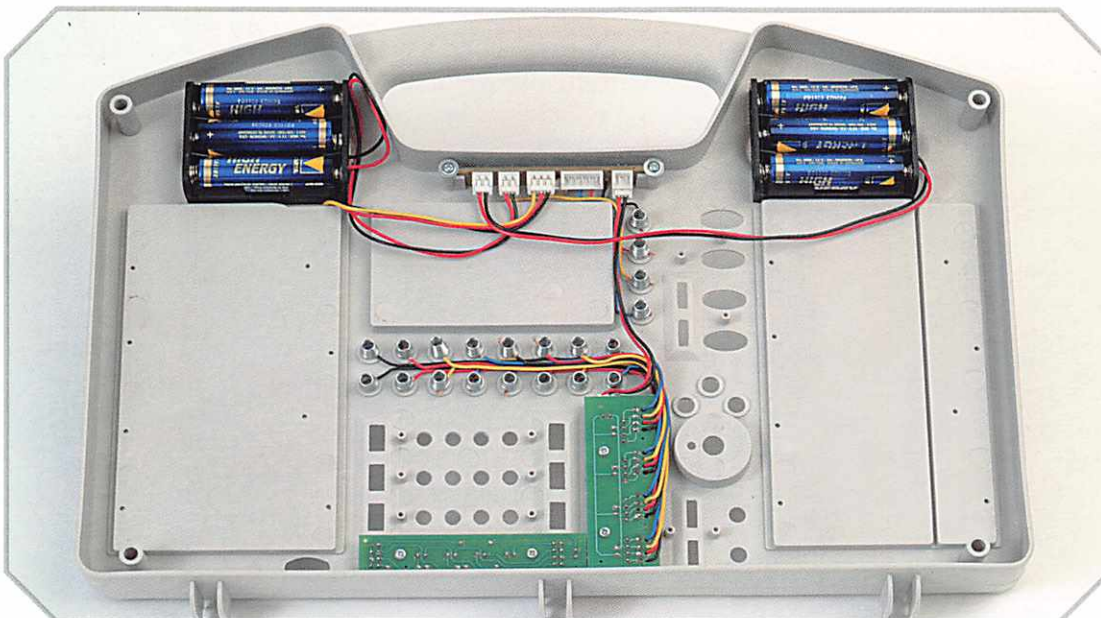
Scheda DG09 con il secondo portabatterie collegato.

J70 per il momento non verrà utilizzato; nel caso in cui sia collegato un portabatterie, è possibile togliere questo connettore separandolo con un cacciavite, applicando quest'ultimo fra il connettore della scheda e il connettore del cavo. Questa scheda riceverà l'alimentazione tramite la scheda DG07, ma fino a quando non disporremo della stessa, utilizzeremo il connettore J69, in modo che sul terminale 1 venga collegato il filo nero e sul 2 il filo rosso. L'altro estremo di questo cavo verrà collegato ai terminali delle molle, siglati come 0 V e 5 V, il nero a 0 V e il rosso a 5 V.

Il secondo portabatterie

Il secondo portabatterie permette di collegare tre pile in più alla scheda di distribuzione dell'alimentazione, in questo modo è possibile disporre di 9 V per il laboratorio. Il collegamento si realizza portando il connettore del portabatterie, situato sotto la zona 2, sino alla scheda DG09 e collegandolo al connettore J94 di questa stessa scheda.

Revisione

È molto importante verificare che il secondo portabatterie sia collegato correttamente al terminale J94. Poniamo l'attenzione su questo fatto perché questa scheda dispone di diversi connettori uguali e potrebbe essere facile commettere qualche errore di collegamento.

Interno del laboratorio con i due portabatterie collegati.

Il condensatore come memoria

In questo esercizio faremo una prova con un circuito molto semplice. Quando si utilizzano delle porte ad alta impedenza di ingresso – che consumano quindi pochissima corrente – se c'è un condensatore collegato sull'ingresso, in mancanza di un circuito di scarica, esso può rimanere carico e memorizzare uno stato logico per molto tempo.

Il circuito

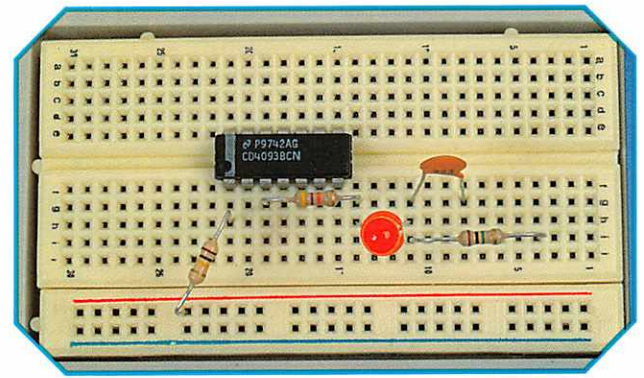
Se osserviamo lo schema ci risulterà molto conosciuto. Abbiamo un circuito integrato 4093, di cui utilizziamo tre porte i cui ingressi sono collegati tra loro in modo che funzionino come porte invertenti. Tra l'uscita della porta U2A e l'ingresso della porta U2C vi è un piccolo circuito formato da due resistenze, R6, R7, il LED A, e un condensatore C1. Questo circuito è l'oggetto dello studio di questa prova.

L'uscita della porta U2C, terminale 10 del circuito integrato, si collega al LED 4, e questa stessa uscita, però invertita, si ottiene sull'uscita della porta U2D, terminale 11 dell'integrato, che utilizza come visualizzatore il LED 3.

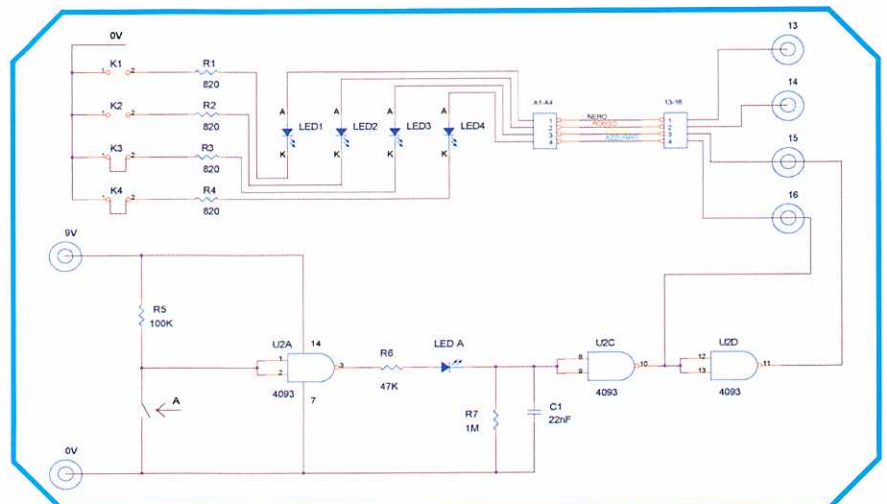
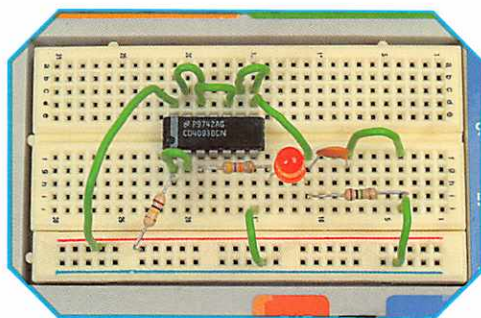
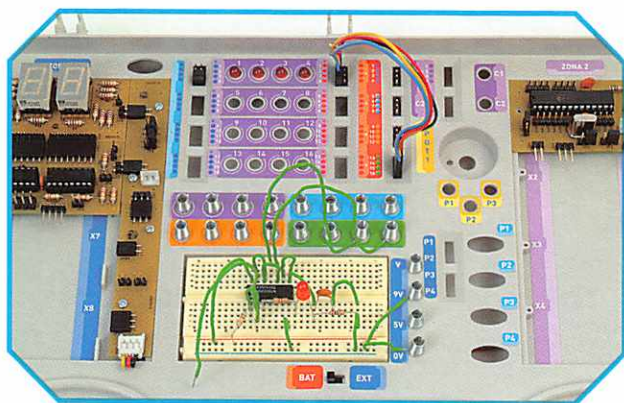
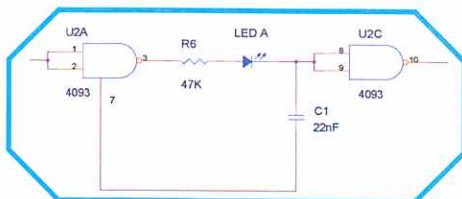
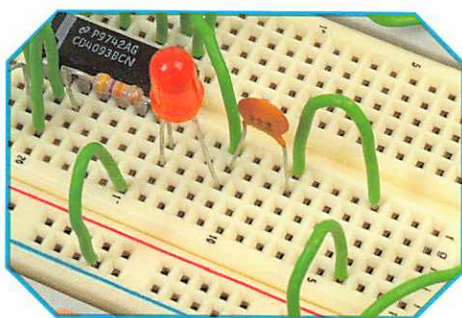
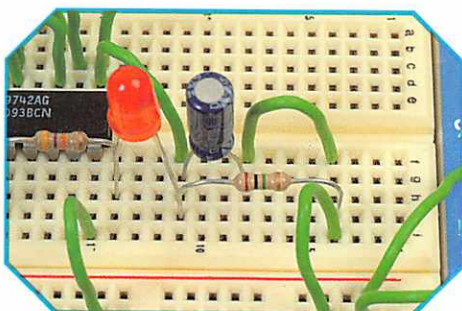
Montaggio

La realizzazione pratica di questo esperimento non comporta alcuna difficoltà, dato che basta montare bene i componenti e realizzare il cablaggio attentamente, rispettando la polarità del LED A. Non dobbiamo dimenticare di collegare i ponticelli tra i catodi dei LED 3 e LED 4, e il negativo dell'alimentazione, ovvero i terminali che si trovano a fianco, dato che senza questi ponticelli i LED non si potrebbero illuminare.

Alimentazione

Dopo aver verificato che il montaggio sia stato eseguito correttamente si collega l'alimentazione, prendendo il negativo dalla molla 0 V, mentre il positivo si può collegare a 5 o 9 V, dato che il 4093 può lavorare perfettamente a entrambe le tensioni. Se si utilizzano i 9 V bisogna collegare le tre pile in entrambi i portabatterie. Il commutatore di alimentazione sarà sulla posizione BAT.

Componenti montati sulla scheda.

Funzionamento

Il funzionamento del circuito è all'inizio molto semplice, però, come vedremo, si può complicare. Se osserviamo lo schema completo, l'ingresso del circuito è a 1 logico, obbligato dalla resistenza di "pull-up" R5, quindi la sua uscita è a livello basso; in questa condizione il LED A non può condurre, quindi l'ingresso

Circuito con condensatore che può memorizzare un dato.

Cablaggio della scheda.

Collegamenti dell'esperimento.

In questo circuito non c'è un percorso di scarica per C1.

Circuito senza la resistenza R7.

In questo caso si aumenta la capacità del condensatore.

della porta U2C è a livello basso, obbligato in questo caso, dalla resistenza R7. L'uscita di questa porta, terminale 10 dell'integrato, è a livello alto, il LED 4 si illumina e il LED 3 rimane spento. Quando chiudiamo il collegamento A in modo permanente, l'uscita della porta U2A passa a livello alto, il LED conduce, e la tensione del condensatore sale in modo rapido quindi sull'ingresso della porta U2C troviamo un livello alto, di conseguenza la sua uscita passa a livello basso, il LED 4 si spegne e il LED 3 si illumina. Fin qui tutto appare molto semplice. Togliamo ora il collegamento A, a questo punto l'ingresso della porta U2A passa a livello alto e la sua uscita a livello basso, però, dato che il condensatore è carico, l'ingresso della porta U2C rimane a livello alto fino a quando il condensatore non si scarica. Questa scarica però, non si può fare tramite la resistenza R6, perché lo impedisce il LED A. Per fare in modo che si produca la scarica è necessario inserire nel circuito la resistenza R7.

Esperimento

Il primo esperimento che si può realizzare consiste nel togliere la resistenza R7. In questo caso, quando chiudiamo il collegamento A, l'uscita della porta U2A passa a livello alto, il condensatore C1 si carica tramite R6 e il LED A, e l'uscita della porta U2C passa a livello basso, spegnendo il LED 4 e illuminando il LED 3, ma questo lo abbiamo visto succedere anche prima.

Se a questo punto apriamo il collegamento A, l'uscita della porta U2A, terminale 10, passa a livello basso e il LED non conduce; in questo caso la resistenza R7 è stata tolta e il condensatore non si può scaricare, pertanto l'uscita di questa porta non cambia ed è necessario collegare nuovamente la resistenza R7 per verificare che il circuito funzioni nuovamente.

Possiamo fare delle prove con capacità maggiori, ad esempio 10 µF, rispettando in questo caso la polarità del condensatore: il terminale positivo si deve collegare ai terminali 8 e 9 dell'integrato; se colleghiamo nuovamente R7 si produce un ritardo maggiore nel circuito.

LISTA DEI COMPONENTI	
U2	Circuito integrato 4093
R5	Resistenza 100 K (marrone, nero, giallo)
R6	Resistenza 47 K (giallo, viola, arancio)
R7	Resistenza 1 M (marrone, nero, verde)
C1	Condensatore 22 nF

Primo esperimento con PIC 16F870

Disponiamo già della scheda DG06 con il PIC programmato. Dato che disponiamo di un programma scritto, che più tardi potremo modificare, scrivere o cancellare, è già possibile utilizzarlo per realizzare qualche esperimento. È arrivato il momento di iniziare l'esecuzione di un progetto in tutte le sue fasi: analisi, progetto, programmazione, simulazione e montaggio.

Il microcontroller PIC 16F870

Il microcontroller PIC 16F870 ha il suo posto riservato nel laboratorio e non sarà più necessario toglierlo da lì per nessuna operazione. Potremo scrivere, cancellare, comunicare con esso mediante un PC e collegarlo a qualsiasi elemento del laboratorio utilizzando cavi di connessione e modificando i "jumper" o "ponticelli" che si trovano sulle schede DG06 e DG07.

Ponticelli della scheda DG06

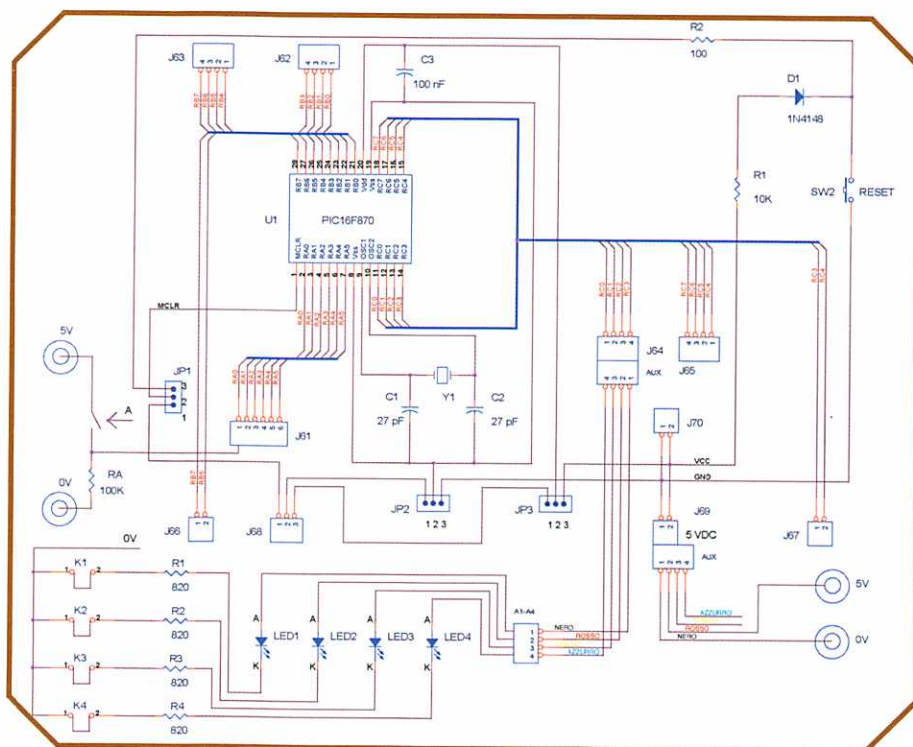
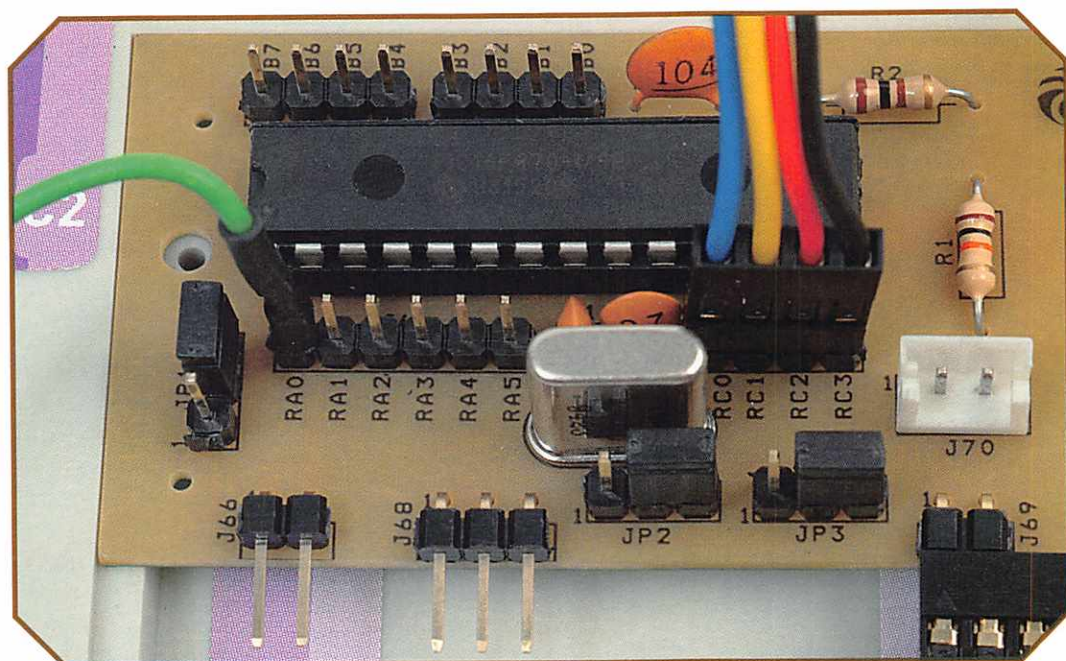
La scheda DG06 ha tre connettori su cui dovremo inserire dei jumper: JP1, JP2 e JP3. Ognuno di questi connettori possiede tre terminali che uniremo mediante un ponticello a due vie; ogni connettore, quindi, ha due combinazioni, in funzione all'inserimento del ponticello sui terminali 1 e 2 oppure su 2 e 3.

I jumper JP2 e JP3

Questi ponticelli selezionano l'origine dell'alimentazione del PIC. Il PIC ha due possibili alimentazioni, una è quella normale del laboratorio che entrerà attraverso il connettore J69 (Vcc e GND), e l'altra è l'alimentazione che utilizzeremo per scrivere il PIC, che entrerà attraverso il connettore J68. Quando programmiamo il PIC prendiamo l'alimentazione direttamente dal PC, infatti, mediante il cavo di comunicazione, questa arriva al laboratorio.

Se guardiamo lo schema elettrico e seguiamo le linee di collegamento vediamo che le linee di massa GND arrivano al JP2 e le linee del positivo o Vcc al JP3.

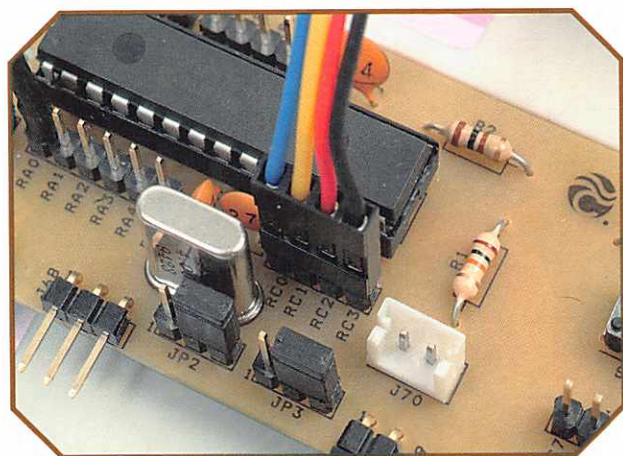
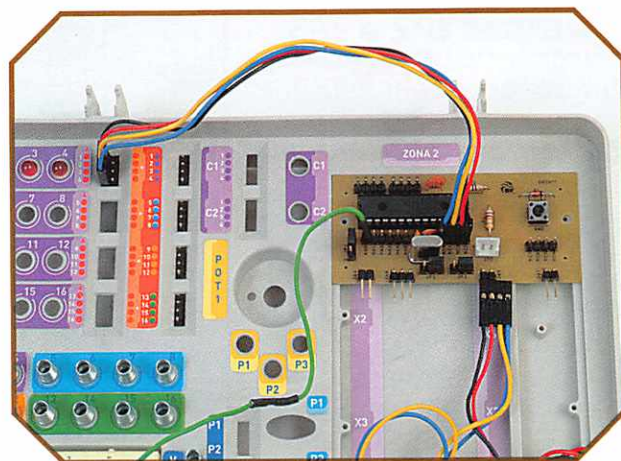
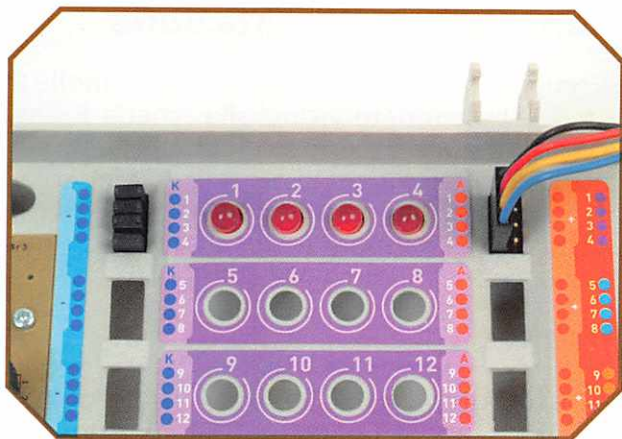
Il terminale 1 sullo schema elettrico è identificato sulla scheda anche con il numero 1, quindi, se vogliamo che l'alimentazione sia quella di funzionamento normale del laboratorio, come nel nostro caso, dobbiamo unire mediante un ponticello i pin 2 e 3 del connettore JP2 e i pin 2 e 3 del connettore JP3.

Osservate che il terminale 2 di questi connettori è quello comune alle possibili combinazioni e va direttamente al microcontroller.

Schema elettrico dell'esperimento.

Scheda DG06 con i ponticelli configurati per il funzionamento normale.

Il ponticello JP1

Questo ponticello ha la funzione di discriminare l'ingresso al terminale MCLR dell'integrato. In questo modo, se stiamo scrivendo il PIC, prenderemo il segnale dal connettore J68 di cui abbiamo parlato in precedenza, altrimenti avremo la possibilità di eseguire il reset mediante un pulsante, come si può vedere nello schema elettrico. Quest'ultima opzione è quella normale di qualsiasi applicazione e, come per i connettori precedenti, per fare in modo che il PIC sia collegato a questo pulsante dovremo inserire il ponticello sui terminali 2 e 3 del JP1. È consigliabile verificare con le fotografie il posizionamento dei ponticelli. Questa è la configurazione normale, e i ponticelli cambieranno posizione solamente nel caso desiderassimo scrivere il PIC.

Enunciato del progetto

Vogliamo realizzare un progetto in cui si generino e si visualizzino una serie di numeri casuali (da 1 a 9 in decimale) che per il momento vedremo in formato binario, utilizzando i LED 1, 2, 3 e 4. Il conteggio si potrà fermare in

Collegamento a RC0-RC3 della porta C.

Gli anodi si collegano a RC0-RC3 con un cavetto a quattro fili.

Sui catodi si collegano i quattro ponticelli.

Montaggio del selettore di ingresso sulla scheda Bread Board.

qualsiasi momento, a quel punto vedremo un numero casuale per tre secondi e, trascorso questo tempo, ricomincerà la visualizzazione sequenziale.

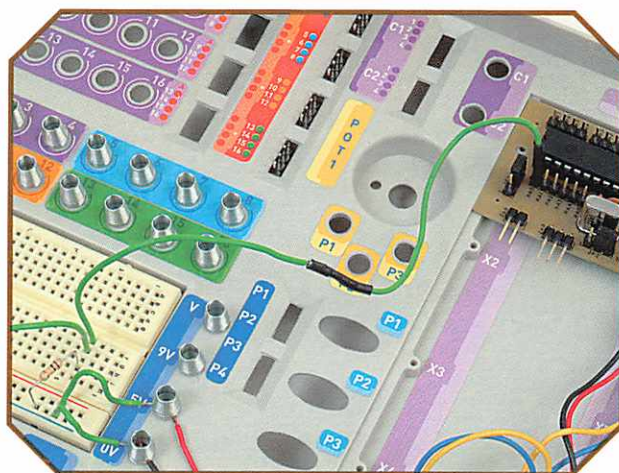
Analisi del progetto

Anche se la sequenza logica, al momento di realizzare un progetto, consiste nel fare prima un'analisi e successivamente realizzare il programma e simularlo, cambiamo l'ordine, realizzando prima un possibile montaggio in modo da poter vedere un risultato finale, dato che il PIC fornito è già programmato con il programma che risolve questo progetto. In seguito spiegheremo nel dettaglio questo programma.

Montaggio: Visualizzazione/Uscite

Utilizzeremo i terminali da RC3 a RC0 della porta C per estrarre il segnale che controlla l'accensione dei LED, i quali rappresentano il numero in binario.

Con uno dei due cavi a 4 fili di cui disponiamo uniremo il connettore J64 (uscite RC0:RC3) della scheda DG06 agli anodi dei diodi LED, come possiamo vedere nello schema e nelle fotografie. Bisogna prestare particolare attenzione nel mantenere una corrispondenza con i colori dei fili. Se incrociamo uno dei cavetti la sequenza risulterà invertita e la visualizzazione non sarà corretta.

Il cavetto di alimentazione si collega a J69.

Collegamento di RA0 alla scheda Bread Board.

Montaggio: Pulsante/Ingresso

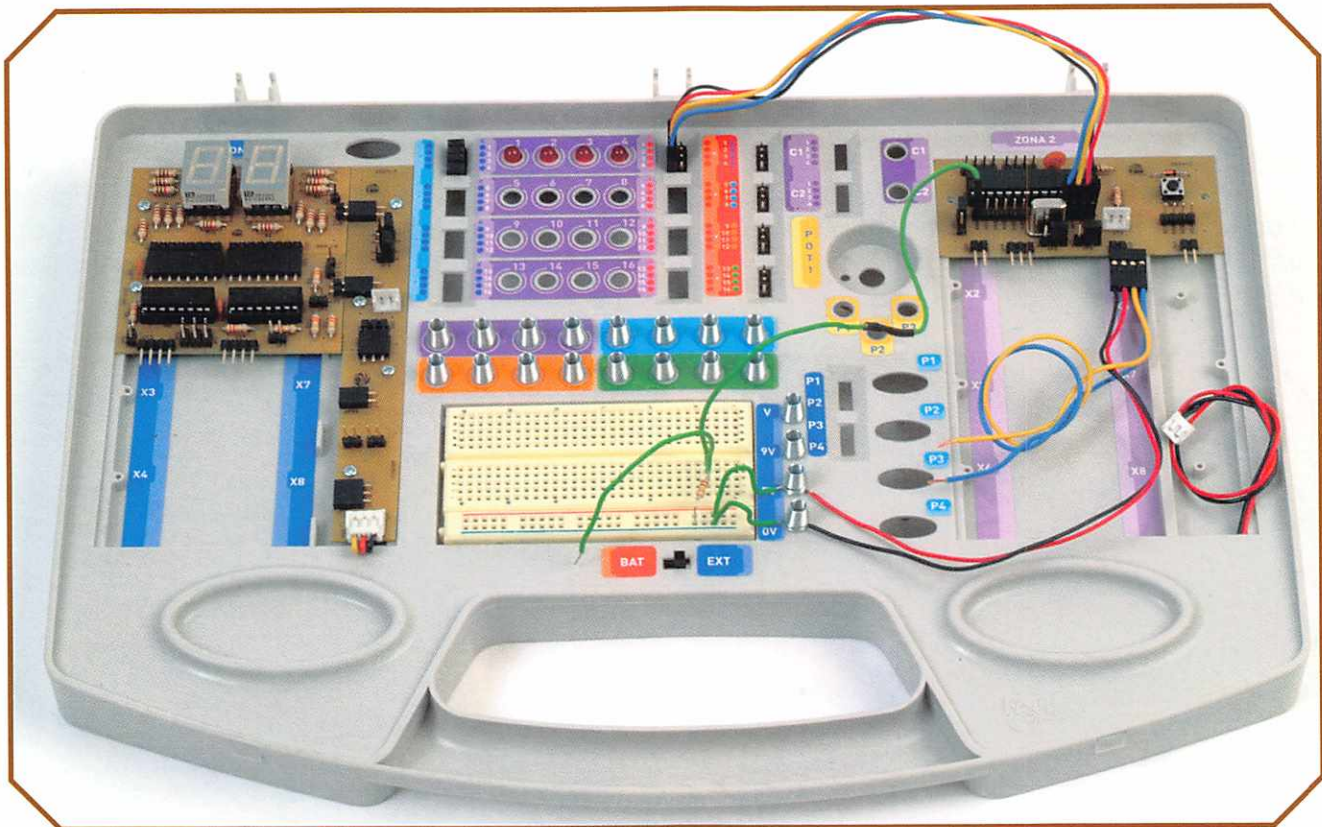
Abbiamo bisogno di un ingresso per indicare al PIC quando deve mostrare il numero casuale. Questo ingresso verrà fornito tramite il pin RA0 della porta A. Quando RA0 è a "1", si vedranno i numeri in modo sequenziale. Quando RA0 torna a livello "0", verrà visualizzato il numero casuale ottenuto.

Dato che per ora non disponiamo dei pulsanti del laboratorio, utilizzeremo un filo per collegare il terminale RA0 al negativo dell'alimentazione, inserendo una resistenza da 100K, resistenza di pull-down, per assicurare uno 0 quando non è attivo il collegamento di controllo, che si esegue sul positivo dell'alimentazione. Questo collegamento di RA0 si realizza con il cavetto a un solo filo prolungato con un ulteriore pezzo di filo di collegamento. Con questo semplice montaggio dobbiamo solamente collegare il cavetto che va da un estremo della resistenza RA da 100 K alla linea del positivo della scheda Bread Board per ottenere un 1, o scollegarlo per ottenere uno 0.

Montaggio: Alimentazione

La scheda DG06 si alimenta con le due molle 0 V e 5 V posizionate vicino alla scheda Bread Board. Per questo scopo impiegheremo un cavetto a quattro fili, di cui ne utilizzeremo solamente due. Il filo rosso verrà utilizzato per il positivo, collegandolo a 5 V, il nero per il negativo, collegandolo a 0 V, come GND.

Risultato

Nella fotografia si può vedere il risultato finale del montaggio. Collegando l'alimentazione al connettore J69 della scheda DG06, con l'interruttore dell'alimentazione in posizione "BAT", ovvero con le batterie, otterremo una visualizzazione sequenziale di numeri con un intervallo di 0,05", e quando agiremo sul terminale RA0 avremo la rappresentazione di un numero casuale per circa 3" approssimativamente. Presto vedremo la visualizzazione diretta in decimale del numero che ora vediamo in binario.

Vista generale dell'esperimento realizzato.

Il repertorio delle istruzioni (IV)

Terminiamo di analizzare il repertorio delle istruzioni del nostro microcontroller con quelle di gestione dei bit e delle istruzioni speciali. Visto questo e facendo pratica con i diversi esempi, non dovremo più aver paura di programmare e saremo capaci di realizzare qualsiasi progetto per complicato che possa apparire.

Istruzioni per la gestione dei bit

Questo insieme di istruzioni ha il compito di verificare il valore di un bit particolare di un registro, di impostarlo a 1 oppure a 0, e anche di produrre un salto di un'istruzione, a seconda se si compie o meno la condizione stabilita.

Nella tabella possiamo vedere le quattro istruzioni che formano questo gruppo.

Impostazione a 0 oppure a 1 di un bit

Le prime due istruzioni servono per l'impostazione a 0 oppure a 1 di un bit di un registro. Un esempio di quanto utile possa risultare avere istruzioni specifiche per questo compito è quando dobbiamo cambiare banco di memoria per lavorare con un registro o scrivere dei dati, o se vogliamo definire un bit di una porta come ingresso o uscita.

BCF f,b: (Bit Clear f) Il bit 'b' contenuto nel registro f si cancella, ovvero si imposta a zero.

BSF f,b: (Bit Set f) Il bit 'b' contenuto nel registro f si setta, ovvero si imposta a uno.

Classificazione delle istruzioni

- Istruzioni di trasferimento
- Istruzioni aritmetiche
- Istruzioni logiche
- Istruzioni di impostazione a zero
- Istruzioni di salto
- Istruzioni per la gestione dei bit
- Istruzioni speciali

Gruppi di istruzioni che formano il repertorio.

Salti dipendenti da un bit

Le due istruzioni successive sono salti dipendenti da un bit. Sono state spiegate nel fascicolo precedente e, anche se i loro nomi possono sembrare complicati da imparare, lo mnemonico stesso dell'istruzione ci indica il loro compito.

Quindi la "b" fa riferimento a un bit, la "t" significa test (verifica di un valore), la "f" è il nome dei registri in assembler, "s" significa salto

MNEMONICO E OPERANDI	DESCRIZIONE	CICLI	CODICE OP	FLAG
BCF f,b	Impostazione a 0 del bit b del registro f	1	01 00bb bfff ffff	——
BSF f,b	Impostazione a 1 del bit b del registro f	1	01 01bb bfff ffff	——
BTFSC f,b	Salta un'istruzione se il bit b del registro f ha valore 0	1 (2)	01 10bb bfff ffff	——
BTFSS f,b	Salta un'istruzione se il bit b del registro f ha valore 1	1 (2)	01 11bb bfff ffff	——

Istruzioni di gestione dei bit.

RP1	RP0	BANCO
0	0	0
0	1	1
1	0	2
1	1	3

Bit RP0 e RP1 del registro STATUS per selezionare il banco di memoria.

(skip), e la terminazione in "c" indicherà 0 (clear) e in "s" 1 (set). La traduzione sarà: "testa il valore di un bit e salta se è 1 (oppure 0)". Mediante queste due istruzioni potremo formare istruzioni di controllo e, grazie alla loro utilità, molto spesso le troveremo implementate nel linguaggio. Negli esempi con cui faremo pratica prossimamente potremo verificarne l'utilità, ma in questo caso, quello dell'allarme di un'automobile, ci serve come esempio veloce. Quando si rileva un'intrusione un bit cambia di stato, normalmente si imposta a 1, e attraverso questa istruzione si determinerà cosa fare nei due stati possibili, come si può vedere nella figura.

BTFSC f,b: (Bit Test, Skip if Clear) Se il bit 'b' del registro 'f' ha valore 1 si esegue l'istruzione successiva. In caso contrario, l'istruzione successiva è saltata e si esegue un'istruzione NOP, facendo quindi in modo che l'istruzione duri due cicli.

BTFSS f,b: (Bit Test, Skip if Set) Come l'istruzione precedente, saltando però l'istruzione successiva nel caso in cui il bit testato sia uguale a 1.

Istruzioni speciali

All'interno di questo gruppo troviamo due istruzioni le cui funzioni sono molto specifiche, quindi non si possono inserire in nessuno dei gruppi precedenti. Le possiamo vedere nella tabella della figura.

NOP: (No Operation) Questa istruzione non ha alcuna funzione. Serve unicamente a far passare un ciclo di istruzione in cui il microprocessore non esegue alcun lavoro.

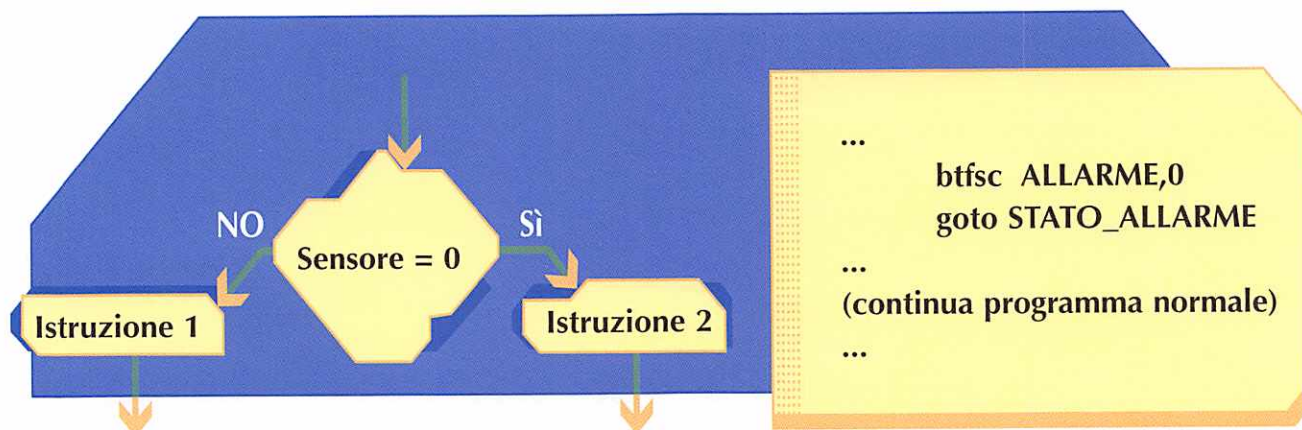
SLEEP: Quando si esegue questa istruzione il microprocessore entra in modo riposo o di basso consumo.

Il bit $\overline{PD}$ (Power-Down) del Registro di Stato si imposta a 0 e il bit $\overline{TO}$ (Time-Out) a 1. Anche il Watchdog e il suo predivisore di frequenza sono impostati a zero.

Facciamo pratica con gli esempi

Abbiamo terminato di vedere il repertorio delle istruzioni del nostro microcontroller, però il miglior sistema per assimilare i nuovi concetti teorici è metterli in pratica, eseguiamo quindi questi diversi esempi.

Esempio 1

Consiste nell'indicare il compito del programma della figura e il tempo di esecuzione impiegato in un PIC 16F870 a 4 MHz.

Esempio di applicazione delle istruzioni di salto in base a un bit con un allarme di automobile.

MNEMONICO E OPERANDI	DESCRIZIONE	CICLI	CODICE OP	FLAG
NOP	Non fa nulla	1	00 0000 0xx0 0000	— —
SLEEP	Porta il processore nel modo riposo o di basso consumo	1	00 0000 0110 0011	$\overline{TO}$, $\overline{PD}$

Istruzioni speciali.

Soluzione: La prima cosa da fare è resettare il registro di lavoro W, impostandolo a 0. Dopo faremo un'operazione OR con il valore binario 10101011 (AB in esadecimale), e lasciamo caricato W con questo valore. Dopodiché eseguiremo un test sul bit 3 del registro W e, dato che vale 1, saltiamo l'istruzione CLRW che avrebbe portato a 0 il registro W, e carichiamo su di esso il valore FFh (11111111 in binario).

Dato che sono cinque istruzioni e una di esse impiega due cicli per essere eseguita, il programma completo, per essere eseguito, impiegherà 6 µs dato che il ciclo di istruzioni è:

$$4 \cdot \frac{1}{4.10^6\ \text{Hz}} = 1\ \mu\text{s}$$

```
CLRW
IORLW     10101011b
BTFSS     W,3
CLRW
MOVLW     FFh
```

Codice dell'Esempio 1.

Esempio 2

Eseguiremo in questo caso un tipico esempio di apprendimento in qualsiasi linguaggio di programmazione. Si tratta di identificare tra due numeri qual è il maggiore. Provate a trovare voi stessi la soluzione senza ricorrere alle figure. Ricordate che, come prima cosa, dovete fare un organigramma che risponda a come risolvere il problema per poi passare a scrivere il codice. Nella figura in basso a sinistra si può vedere una delle possibili soluzioni dell'enunciato proposto.

Per simulare il programma dobbiamo ricorrere alla finestra Modify e inserire valori nelle variabili da comparare. Se in VAR1 inseriremo un 9 e in VAR2 inseriremo un 8, eseguendo il programma vedremo che la sua esecuzione si ferma nel ciclo MINORE, dato che VAR2<VAR1. Se cambiamo il valore di VAR2 inserendo una A (10 in decimale), vedremo che il programma si ferma nel ciclo maggiore.

Nella figura della pagina precedente possiamo vedere la videata di simulazione utilizzata in questo programma.

```
c:\progra~1\mplab\progetti\mag_min.asm

            list     p=16F870
            include  "p16F870.inc"

VAR1        EQU      21
VAR2        EQU      22

            ORG      0              ; Indirizzo di Reset
            goto     INIZIO

INIZIO      movf     VAR1,W         ; Sposta il valore di VAR1 su W
            subwf    VAR2,W         ; Sottrae VAR2-W (VAR1) e lascia il risultato su W
            btfss    STATUS,Z       ; Verifica il valore del bit Z
            goto     DIVERSI        ; Istruzione se sono diversi
            goto     UGUALI         ; Istruzione se sono uguali

DIVERSI     btfss    STATUS,C       ; Verifica il valore del bit C
            goto     MINORE         ; VAR2<VAR1
            goto     MAGGIORE       ; VAR2>VAR1

UGUALI      goto     UGUALI
MINORE      goto     MINORE
MAGGIORE    goto     MAGGIORE

            end
```

Codice dell'Esempio 2.

Simulazione dell'Esempio 2. Sarà necessario utilizzare la finestra Modify per inserire i valori da comparare.

RA1	RA0	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	1	0	1
1	0	1	0	1	0	1	0	1	0
1	1	1	1	1	1	1	1	1	1

Sequenza di uscita della porta B nell'Esempio 3.

Esempio 3

In quest'ultimo esempio vogliamo imparare a utilizzare le strutture di controllo e iniziare ad avvicinare i nostri programmi ad applicazioni più reali. In questo esercizio di tipo combinazionale si tratta di fornire diverse combinazioni di uscita in funzione dello stato logico dei due segnali di ingresso.

In base allo stato dei segnali collegati ai pin RA0 e RA1, si attivano le uscite collegate alla porta B (RB0:RB7) conforme alla tabella della verità della figura.

Supponiamo di realizzare un semplice controllo di illuminazione mediante due interruttori.

Quando nessun interruttore è premuto tutte le luci rimangono spente, quando si preme uno di essi si accenderanno le zone gestite da questo interruttore, e quando li attiveremo tutti e due insieme, tutte le luci si accenderanno. Questa è una delle possibili applicazioni che si possono fare con questo esercizio.

Quando simuleremo questo esempio potremo verificare solamente una combinazione RA0=RA1=0, in quanto questi pin non potranno essere forzati su valori diversi. Quello che faremo sarà caricare sul microcontroller il programma, dopo aver imparato a programmarlo, e lo verificheremo col suo hardware corrispondente (due interruttori o, in mancanza, due fili, e otto diodi LED con le loro resistenze di limitazione collegate alla porta B).

c:\programmi\mplab\progetti\ed27_3.asm

```
        list    p=16F870        ;Tipo di processore
        include "P16F870.INC"   ;Definizione dei registri interni

        org     0x00            ;Vector di reset
        goto    Inizio

        org     0x05            ;Salva il vector di interrupt

Inizio  clrf    PORTB           ;Cancella il latch di uscita
        bsf     STATUS,RP0      ;Seleziona il banco 1
        clrf    TRISB           ;Porta B configurata come uscita
        movlw   b'00011111'
        movwf   TRISA           ;Porta A configurata come ingresso
        bcf     STATUS,RP0      ;Seleziona il banco 0

Ciclo   clrwdt                  ;Aggiorna il WDT
        btfsc   PORTA,0         ;Verifica lo stato di RA0
        goto    RA0_è_1         ;Se è 1 salta a questa etichetta
        btfsc   PORTA,1         ;RA0=0 e testa lo stato di RA1
        goto    Sono_10         ;Salta a questa etichetta se RA1=1
        clrf    PORTB           ;Sono a 00, uscite tutte a zero
        goto    Ciclo

Sono_10 movlw   b'10101010'
        movwf   PORTB           ;Uscita per la combinazione 10
        goto    Ciclo

RA0_è_1 btfsc   PORTA,1         ;Verifica lo stato di RA1
        goto    Sono_11         ;Salta se RA1=1
        movlw   b'01010101'
        movwf   PORTB           ;Uscita per la combinazione 01
        goto    Ciclo

Sono_11 movlw   b'11111111'
        movwf   PORTB           ;Uscita per la combinazione 11
        goto    Ciclo

        end                     ;Fine del programma sorgente
```

Codice che risolve l'Esempio 3.